UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,800	10/27/2003	Hideo Miyake	1450.1005D	1082

21171 7590 12/27/2007

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
GEIB, BENJAMIN P	

ART UNIT	PAPER NUMBER
2181	

MAIL DATE	DELIVERY MODE
12/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/692,800	MIYAKE ET AL.	
	Examiner	Art Unit	
	Benjamin P. Geib	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 October 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *14-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *14-34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. Applicant, via amendment, has overcome the 35 U.S.C. § 112, second paragraph, rejections set forth in the previous Office Action. Consequently, the examiner has withdrawn these rejections.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 14-20, 33, and 34 are rejected under 35 U.S.C. 102(b) as being anticipated by Alpert et al., U.S. Patent No. 5,740,413 (Hereinafter Alpert).

4. Referring to claim 14, Alpert has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated branch condition of the conditional instruction is satisfied, wherein a determination of the branch condition of the conditional instruction and the executed data processing when the branch condition of the conditional instruction is satisfied are indivisible, said apparatus comprising:

a break detection section for detecting a breakpoint set at an arbitrary position of an instruction sequence *[detecting a branch instruction when the branch breakpoints are enabled; column 6, lines 38-55]*;

a condition determination section for determining whether a branch condition of said conditional instruction is satisfied *[determining if the branch is taken; column 6, lines 38-55]*; and

a control section for controlling a break-interrupt based upon of a breakpoint detection result from said break detection section and a branch condition determination result from said condition determination section *[a breakpoint is generated if a branch is taken and branch breakpoints are enabled; column 6, lines 38-55]*.

5. Referring to claim 15, Alpert has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated branch condition of the conditional instruction is satisfied, wherein a determination of the branch condition of the conditional instruction and the executed data processing when the branch condition of the conditional instruction is satisfied are indivisible, said apparatus comprising:

an instruction break detection section for detecting an instruction break in accordance with whether an instruction corresponding to an instruction address representing a breakpoint, which is set in a register, is read, and outputting a detection signal representing a detection result *[detecting a breakpoint using the address breakpoint unit; column 6, line 63 – column 7, line 10]*;

a condition determination section for determining whether a branch condition the conditional instruction is satisfied, and outputting a branch condition determination signal *[the branch breakpoint unit sends a signal in response to determining that branch is or will be taken; column 6, lines 38-55]*; and

a logical operation section for performing AND operation to said detection signal output from said instruction break detection section and said branch condition determination signal output from said condition determination section, and sending a break-interrupt notification in accordance with the AND operation result *[the debug circuitry generates a breakpoint in accordance with the address breakpoint unit and branch breakpoint unit; column 6, lines 20-31]*.

6. Referring to claim 16, Alpert has taught an apparatus according to claim 15, wherein

said condition determination section is designed to determine whether an instruction word is said conditional instruction, if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal *[column 6, lines 38-55]*, and

when an instruction word corresponding to the instruction address representing said breakpoint is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, said logical operation section does not send the break-interrupt notification, and when said instruction word is the conditional instruction having a satisfied branch condition, said logical operation section sends said break-interrupt notification *[column 6, lines 38-55]*.

7. Referring to claim 17, Alpert has taught an apparatus according to claim 15, wherein

said apparatus further comprises a mode setting section for setting one of a first mode in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the branch condition of said conditional instruction is satisfied, and a second mode in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[enabling/disabling the enable bit for the branch breakpoint unit changes between a modes that generates a breakpoint on a taken branch and one that doesn't; column 6, lines 20-31]*,

said condition determination section is designed to determine whether an instruction word is said conditional instruction, if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal *[column 6, lines 38-55]*, and

in said first mode, when an instruction word corresponding to the instruction address representing said breakpoint is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, said logical operation section does not sends the break-interrupt notification, and when said instruction word is the conditional instruction having a satisfied branch condition, said logical operation section sends the break-interrupt notification, and in said second mode, when said instruction word is an instruction word corresponding to the instruction address

representing said breakpoint, said logical operation section sends said break-interrupt notification *[column 6, lines 20-31]*.

8. Referring to claim 18, Alpert has taught an apparatus according to claim 15, wherein

said condition determination section is designed to determine whether an instruction word is said conditional instruction, if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal *[column 6, lines 38-55]*, and

when an instruction word corresponding to the instruction address representing said breakpoint is a conditional instruction having an unsatisfied branch condition, said logical operation section does not send the break-interrupt notification, and when said instruction word is an unconditional instruction or the conditional instruction having a satisfied branch condition, said logical operation section sends said break-interrupt notification *[column 6, lines 38-55]*.

9. Referring to claim 19, Alpert has taught an apparatus according to claim 15, wherein

said apparatus further comprises a mode setting section for setting one of a first mode in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the branch condition of said conditional instruction is satisfied, and a second mode in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[enabling/disabling the enable bit for the branch breakpoint unit changes between a modes that generates a breakpoint on a taken branch and one that doesn't; column 6, lines 20-31]*,

said condition determination section is designed to determine whether an instruction word is said conditional instruction, if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal *[column 6, lines 38-55]*, and

in said first mode, when an instruction word corresponding to the instruction address representing said breakpoint is a conditional instruction having an unsatisfied branch condition, said logical operation section does not send the break-interrupt notification, and when said

instruction word is an unconditional instruction or the conditional instruction having a satisfied branch condition, said logical operation section sends the break-interrupt notification, and in said second mode, when said instruction word is an instruction word corresponding to the instruction address representing said breakpoint, said logical operation section sends said break-interrupt notification *[column 6, lines 20-31]*.

10. Referring to claim 20, Alpert has taught an apparatus according to claim 15, wherein said data processing system comprises one of a scalar processor for performing one unit of processing in accordance with one instruction, a long instruction word processor for parallel executing short instructions forming a long instruction word, and a parallel processor for parallel executing at least one basic instruction forming a variable-length instruction word *[column 5, lines 14-16]*.

11. Referring to claim 33, Alpert has taught an interrupt control method for controlling a break-interrupt in a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated branch condition of the conditional instruction is satisfied, wherein a determination of the branch condition of the conditional instruction and the executed data processing when the branch condition of the conditional instruction is satisfied are indivisible, said method comprising the steps of:

detecting a breakpoint set at an arbitrary position of an instruction sequence *[detecting a branch instruction when the branch breakpoints are enabled; column 6, lines 38-55]*;

determining by a branch condition determining section whether a condition of a branch condition said conditional instruction is satisfied *[determining if the branch is taken; column 6, lines 38-55]*; and

controlling the break-interrupt based upon the detecting of said breakpoint and the determining of the branch condition of said conditional instruction from the branch condition determining section *[a breakpoint is generated if a branch is taken and branch breakpoints are enabled; column 6, lines 38-55]*.

12. Referring to claim 34, Alpert has taught an apparatus comprising:

A controller *[branch breakpoint unit; FIG. 1, component 190]*

detecting a breakpoint set at an arbitrary position of an instruction sequence *[detecting a branch instruction when the branch breakpoints are enabled; column 6, lines 38-55]*;

determining a branch of an instruction *[determining if the branch is taken; column 6, lines 38-55]*; and

controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction, according to a logical operation of a detection signal from said breakpoint detection and a branch condition determination signal from said branch control determination of the instruction *[the branch breakpoint unit determines if a branch is taken and branch breakpoints are enabled; column 6, lines 38-55]*; and

sending a break-interrupt notification in accordance with the logical operation *[a breakpoint is generated if a branch is taken and branch breakpoints are enabled; column 6, lines 38-55]*.

13. Claims 21-32 are rejected under 35 U.S.C. 102(e) as being anticipated by Alverson et al., U.S. Patent No. 6,480,818 (Herein referred to as Alverson).

14. Referring to claim 21, Alverson has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated branch condition of the conditional instruction is satisfied, wherein a determination of the branch condition of the conditional instruction and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

an instruction break detection section *(target thread execution subroutine; Fig. 11, component 1100)* for detecting an instruction break in accordance with whether an instruction corresponding to an instruction address representing a breakpoint, which is set in a register, is

read *[The break instruction, which corresponds to an address and is set in a register (See Fig. 4B), is inherently read out for instruction execution]*, and sending a break-interrupt notification in accordance with the detecting of the instruction break *[The target thread execution subroutine detects an instruction break (i.e. breakpoint; See Fig. 11, component 1110) and notifies the breakpoint handler subroutine (i.e. sends a break-interrupt notification); See column 21, lines 15-39 and Fig. 11]*; and

a control section *(breakpoint handler subroutine; Fig. 12, component 1125)* for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said instruction break detection section *(target thread execution subroutine)*, determining whether a branch condition of said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with the determining of the branch condition of the conditional instruction *[The breakpoint handler subroutine determines if a condition of the conditional instruction is satisfied (and, therefore, the breakpoint is valid) and notifies the nub (i.e. controls break-interrupt processing) if the condition is satisfied; See column 21, lines 51-66].*

15. Referring to claim 22, Alverson has taught an apparatus according to claim 21, wherein said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction *(The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210)*, and when said instruction word is said conditional instruction, determines whether the branch condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is a conditional instruction having a satisfied branch condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

16. Referring to claims 23 and 29, taking claim 23 as exemplary, Alverson has taught an apparatus according to claim 21, wherein

said apparatus further comprises a mode setting section *(nub thread execution routine; Fig. 5, component 500)* for setting one of a first mode *(mode when breakpoint set is a conditional breakpoint)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of the branch of said conditional instruction is satisfied, and a second mode *(mode when breakpoint set is an unconditional breakpoint)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[The nub thread execution routine sets a breakpoint mode by recording information indicating whether the inserted breakpoint is conditional or not; column 15, lines 43-63]*, and

in said first mode, said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction *(See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether the condition of the branch of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is a conditional instruction having a satisfied branch condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*, and

in said second mode, said control section performs said break-interrupt processing when receiving said break-interrupt notification *[In the second mode the breakpoint set is unconditional and, therefore, the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*.

17. Referring to claims 24 and 30, taking claim 24 as exemplary, Alverson has taught an apparatus according to claim 21, wherein said control section determines, in said interrupt

handler, whether an instruction word as an instruction break target is said conditional instruction *(The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether the condition of the branch of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having a satisfied branch condition, performs said break-interrupt processing *[When the breakpoint is unconditional or conditional and the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

18. Referring to claims 25 and 31, taking claim 25 as exemplary, Alverson has taught an apparatus according to claim 21, wherein

said apparatus further comprises a mode setting section *(nub thread execution routine; Fig. 5, component 500)* for setting one of a first mode *(mode when breakpoint set is a conditional breakpoint)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of the branch of said conditional instruction is satisfied, and a second mode *(mode when breakpoint set is an unconditional breakpoint)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[The nub thread execution routine sets a breakpoint mode by recording information indicating whether the inserted breakpoint is conditional or not; column 15, lines 43-63]*, and

in said first mode, said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction *(See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether the condition of the branch of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is a conditional instruction

having an unsatisfied branch condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having a satisfied branch condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) or breakpoint is unconditional the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65]*, and

in said second mode, said control section performs said break-interrupt processing when receiving said break-interrupt notification *[In the second mode the breakpoint set is unconditional and, therefore, the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

19. Referring to claims 26 and 32, taking claim 26 as exemplary, Alverson has taught an apparatus according to claim 21, wherein said data processing system comprises one of a scalar processor for performing one unit of processing in accordance with one instruction *(See Fig. 3, component 101 and column 1, lines 20-40)*, a long instruction word processor for parallel executing short instructions forming a long instruction word, and a parallel processor for parallel executing at least one basic instruction forming a variable-length instruction word.

20. Referring to claim 27, Alverson has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated branch condition of the conditional instruction is satisfied, wherein a determination of the branch condition of the conditional instruction and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

a software break detection section *(target thread execution subroutine; Fig. 11, component 1100)* for detecting a software break in accordance with whether a breakpoint instruction placed at an arbitrary position of an instruction sequence is executed *(See Fig. 4)*, and sending a break-interrupt notification in accordance with the detection of the software break *[The*

target thread execution subroutine detects a software break (i.e. breakpoint; See Fig. 11, component 1110) and notifies the breakpoint handler subroutine (i.e. sends a break-interrupt notification); See column 21, lines 15-39 and Fig. 11]; and

a control section *(breakpoint handler subroutine; Fig. 12, component 1125)* for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said software break detection section *(target thread execution subroutine)*, determining whether a branch condition of said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with the determining of the branch condition of the conditional instruction *[The breakpoint handler subroutine determines if a condition of the conditional instruction is satisfied (and, therefore, the breakpoint is valid) and notifies the nub (i.e. controls break-interrupt processing) if the condition is satisfied; See column 21, lines 51-66].*

21. Referring to claim 28, Alverson has taught an apparatus according to claim 27, wherein said control section determines, in said interrupt handler, whether an instruction word as a software break target is said conditional instruction *(The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210)*, and when said instruction word is said conditional instruction, determines whether the branch condition of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, and when said instruction word as said software break target is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler *(The breakpoint handler subroutine always eventually returns; See last step of Fig. 12)*, and when said instruction word as said software break target is a conditional instruction having a satisfied branch condition, performs said break-interrupt processing *[When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

Response to Arguments

22. Applicant's arguments filed 10/16/2007 have been fully considered but they are not persuasive.

23. Applicant argues the novelty/rejection of claims 21-32 and 34 on pages 13-14, in substance that:

> "However, in Alverson a conditional breakpoint merely refers to whether the nub has designated the breakpoint as valid or invalid"
>
> "Alpert cannot anticipate the claimed embodiment by failing to disclose or inherently 'determine a branch of an instruction; and controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction' and 'controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction, according to a logical operation of a detection signal from said breakpoint detection and a branch condition determination signal from said branch condition determination of the instruction; and sending a break-interrupt notification in accordance with the logical operation.'"

These arguments are not found persuasive for the following reasons:

Regarding the applicant's argument that the conditional breakpoint of Alverson "merely refers to whether the nub has designated the breakpoint as valid or invalid", the examiner notes that determining whether a breakpoint is valid or invalid is "determining whether a branch condition of said conditional instruction is satisfied" as claimed. It appears to the examiner that the applicant intends for this condition to be read as a more specific type of condition. If such is the case, the claims should be amended to require such a reading.

Regarding applicant's argument with respect to Alpert, the examiner notes that Alpert has taught generating a breakpoint when a taken branch is encountered and branch breakpoints are enabled [Alpert; column 6, lines 38-55]. Therefore, Alpert has taught determining "a branch of an instruction; and controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction" and "controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction, according to a logical operation of a detection signal from said breakpoint detection and a branch condition determination

signal from said branch condition determination of the instruction; and sending a break-interrupt notification in accordance with the logical operation." In response to applicant's assertion that Alpert does not discuss the detail the branch breakpoint unit, the examiner refers the applicant to U.S. Patent No. 5,659,679 (cross-referenced by Alpert), which further details the branch breakpoint unit.

24. Applicant's arguments with respect to claims 14-20 and 33 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

25. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin P. Geib whose telephone number is (571) 272-8628. The examiner can normally be reached on Mon-Fri 8:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alford Kindred can be reached on (571) 272-4037. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Benjamin P Geib
Examiner
Art Unit 2181

ALFORD KINDRED
SUPERVISORY PATENT EXAMINER